

BAB II

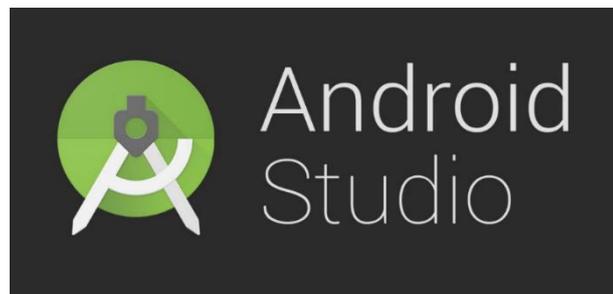
LANDASAN TEORI

Berikut ini adalah beberapa tinjauan pustaka yang menjadi pokok bahasan dalam pengerjaan tugas akhir:

2.1. Android Studio

Android studio adalah *tool* resmi untuk pengembangan secara terintegrasi. Aplikasi Android studio sifatnya adalah *open source* atau bisa dibilang gratis dan dibuat dengan bersumber pada *IDE java* yang biasa digunakan dengan sebutan IDEA IntelliJ [1]. Selain itu, Android Studio juga terdapat memiliki fitur:

- a. Menggunakan *gradle build* yang fleksibel
- b. Dapat membuild Aplikasi dengan mudah
- c. Kompatibel dengan beberapa servis google dan kompatibel dengan berbagai perangkat android.
- d. Penggunaan *edit* di dalam aplikasi yang menarik.



Gambar 2. 1 Android Studio

2.2. Google Maps API

Google Maps merupakan fasilitas peta gratis yang disediakan oleh perusahaan Google yang begitu populer. Google Maps merupakan peta digital yang dapat dengan bebas digunakan dalam mencari suatu daerah dengan lengkap, data peta yang diambil menggunakan satelit. Selain itu pengguna juga dapat menggunakan fitur dengan cara menambahkan Google Maps didalam website yang telah pengguna buat atau mungkin pengguna juga bisa menambahkan pada *blog* pribadi seperti *wordpress*. Penggunaan google maps api ini ada yang berbayar ataupun gratis. Sebagai contoh penggunaan API Google Maps adalah memberi *marker* pada peta android, penentuan jarak antara suatu titik menggunakan *direction* API dan lain-lain. [6]



Gambar 2. 2 Logo Google Maps

2.3. Sistem Operasi Android

Android adalah sebuah sistem yang dibuat oleh perusahaan raksasa google dan sifatnya bersifat *open source* artinya siapa saja bisa memodifikasi yang ada didalamnya. Perangkat yang memakai sistem operasi Android dapat menggunakan untuk piranti yang seperti contohnya pada ponsel pintar lainnya dan juga beberapa tablet yang berbasis android. Selain itu juga android sangat populer dikarenakan android akan mudah dijumpai dan sangat fleksibel untuk digunakan. Selain itu sistem operasi android juga memiliki fitur lain seperti *multi touch* yang berarti ponsel android bisa melakukan klik dalam beberapa titik pada layar, kemudian adalah browser pada android terintegrasi dan memiliki basis *open source webkit* [10].



Gambar 2. 3 Logo Android

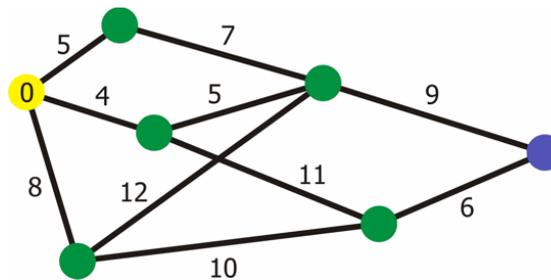
2.4. Algoritme *Dijkstra*

Algoritme *dijkstra* adalah sebuah algoritme yang dipakai dalam mencari rute. Algoritme *Dijkstra* juga dipakai untuk menuntaskan mencari lintasan terpendek [2]. Persolan tentang rute yang terpendek dari suatu tempat ke tempat lain adalah sebuah masalah yang sangat banyak dipergunakan untuk percobaan apakah sebuah algoritme yang digunakan berhasil atau tidaknya. Untuk persoalan jarak yang terpendek mungkin cukup bagus dalam mewakili masalah kecepatan, oleh karena itu masalahnya jadi mudah dipahami dengan cara menjumlahkan seluruh *edge*. namun dengan demikian *Dijkstra* juga mempunyai banyak pilihan solusi. Berikut adalah deskripsi matematis untuk menjumlahkan graf yang dapat digantikan $G = \{V, E\}$ dengan kata lain sebuah graf G dideskripsikan oleh simpul *vertex* dan *edge*. Berikut adalah tahapan alur logika dari algoritme *Dijkstra*:

1. beri nilai (jarak) untuk dari satu titik ke titik lainnya, setelah itu berikan nilai kosong (0) di *node* awal dan berikan *node* tak hingga di *node* lain yang masih belum di lewati.
2. Langkah selanjutnya adalah memberi tanda “belum dilewati” dan berikan *node* awal sebagai *node* awal.
3. Selanjutnya adalah dari *node* awal, nilai *node* lain yang belum dilewati dan kemudian kita hitung jarak dari titik awal. Sebagai contohnya adalah titik berangkat dari E-F memiliki bobot 2 dan dari F-G memiliki jarak 4 maka dari itu dari titik E ke titik G dengan melewati titik F didapatkan hasil $2+4=6$. Jika data jarak yang barusan di hitung lebih kecil daripada jarak sebelumnya

maka kita harus menghapus data lama dan kita dapat menyimpan dengan data jarak yang lebih terbaru yang telah dihitung.

- Langkah selanjutnya adalah set *node* belum dilewati dari *node* awal tadi.



Gambar 2. 4 Gambar Algoritme Dijkstra

2.5. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek adalah pemrograman yang berisi suatu data atau yang biasa disebut sebagai atribut dan juga fungsi yang biasa dikenal sebagai *method*. Konsep dalam pemrograman berorientasi objek adalah:

- Kelas (*class*)

Kelas adalah penggambaran dari set objek yang mempunyai atribut yang serupa. Kelas juga sama dengan tipe data yang terdapat di pemrograman non objek dan tentu saja sifatnya lebih komprehensif karena memiliki ciri karakteristik.

- Objek

Objek memiliki data & fungsi yang sudah terbungkus menjadi 1 unit di dalam program komputer. Selain itu objek juga sebagai dasar dari struktur dalam program yang berorientasi objek.

- Abstraksi

Abstraksi adalah suatu kemampuan dalam sebuah program dalam melewati suatu informasi yang diperoleh. Suatu objek di dalam sistem dapat memiliki berbagai bentuk model.

4. Enkapsulasi

Enkapsulasi adalah proses yang akan memastikan apakah pengguna objek tidak mewakilkan keadaan dari bentuk objek yang berarti hanya objek terstruktur saja yang akan di izinkan untuk dapat mengakses keadaan tertentu yang diinginkan.

5. Polimorfisme

Polimorfisme adalah fungsi yang dimaksudkan dengan cara yang berbeda. Pada pemrograman berorientasi objek, pengguna yang membuat suatu program akan dapat berbagai cara untuk mengimplementasikan suatu fungsi.

6. Inheritas (Pewarisan)

Inheritas adalah sebuah konsep yang dapat mengatur fungsi polimorfisme dan juga enkapsulasi. Objek tersebut dapat mengembangkan perilaku.

2.6 *Unified Modeling Language (UML)*

UML atau dengan kata lain adalah bahasa pemodelan terpadu adalah suatu cara pemodelan dengan secara visual yang di peruntukan dalam membuat suatu perangkat lunak yang berorientasi objek (*object oriented*) [5]. Merancang suatu model objek dari sistem yang begitu kompleks, dan bahkan begitu penting, Semakin kompleks sistem yang ada, semakin penting juga pemanfaatan teknik pemodelan yang baik juga [4]. Selain itu UML dapat memiliki penulisan kelas, dan Bahasa program tertentu, kemudian adalah skema di dalam database. Ada beberapa diagram di dalam unified modelling language antara lain sebagai berikut:

1. *Use Case Diagram*

Use case diagram adalah sebuah *diagram* yang memiliki relasi antara aktor dan *use case*. *Use case* diagram biasanya digunakan untuk pembuatan desain sistem.

2. *Class Diagram*

Class diagram adalah suatu *diagram* yang memiliki cara untuk menunjukkan relasi antar kelas yang terdapat fungsi serta atribut dari objek.

3. *Activity Diagram*

Activity diagram adalah suatu *diagram* yang menjelaskan bagaimana cara data control bekerja serta terstruktur. *Activity diagram* ini terancang dengan rapih di sistem.

4. *Sequence Diagram*

Sequence Diagram adalah suatu *diagram* yang menjelaskan tentang penggabungan dari semua objek yang saling berinteraksi.

2.7 Lintasan Terpendek

lintasan terpendek adalah persoalan untuk mendapatkan dan mencari lintasan pada 2 atau lebih simpul. persoalan ini juga merupakan persoalan optimasi dengan memakai graf berbobot. persoalan ini dapat menentukan jarak antar titik dan sebagainya [9]. Ada beberapa macam persoalan lintasan terpendek, antara lain:

- a. Lintasan terpendek antara dua buah *verteks*.
- b. Lintasan terpendek antara semua pasangan *verteks*.
- c. Lintasan terpendek dari *verteks* tertentu ke semua *verteks* yang lain
- d. Lintasan terpendek antara dua buah *verteks* yang melalui beberapa *verteks* tertentu.

2.8 SQLite

Sqlite adalah sebuah basis data *OpenSource* yang dikhususkan untuk pengembangan aplikasi android karena *database* ini langsung tertanam pada aplikasi. Selain itu SQLite juga tidak mempunyai *server* yang dipisah dari aplikasi android, dengan kata lain sqlite adalah *database* yang tertanam di ponsel. SQLite terdapat pada semua piranti android. Dengan menggunakan perintah SQLite *create* *update* dan *delete* akan mempermudah menggunakan *database* [5].



Gambar 2. 5 SQLite Database

2.9 Sistem Informasi Geografis

Sistem Informasi Geografis adalah sebuah sarana informasi yang terdapat didalam *software* yang berbasis komputer dan dibuat untuk simpan, mengumpulkan dan melakukan olah data, menganalisa data yang akan dicari dan mendapatkan suatu tampilan data dari sebagian objek yang mungkin berhubungan dengan tata letak gambaran di bumi. Suatu elemen SIG antara lain adalah perangkat lunak serta perangkat keras, dan data pengguna aplikasi [3].

2.10 *Location Based Services*

Location Based Services (LBS) adalah layanan berbasis lokasi yang dipergunakan untuk menemukan lokasi perangkat yang sedang digunakan [7]. LBS juga merupakan implementasi dari *GIS Mobile* fungsinya yaitu menampilkan objek seperti navigasi, kota, pencarian suatu alamat. Layanan informasi *Location Based Services* ini dapat memberi interaksi 2 arah antar pengguna [8].